Normalization in DBMS: FastBite Fast Food Ordering System Copy

SCENARIO

FastBite is a growing fast-food chain. They need a clean and well-structured database to manage orders, products, employees, and branches. You have been hired as part of the database design team to improve their current messy data.

Your task: **Normalize the database step-by-step** following the principles of 1NF to 5NF.

ACTIVITY FLOW

Stage 1 – Messy Menu → First Normal Form (1NF)

Goal: Ensure all data is atomic (no multiple values in one cell).

Instructions:

1. Below is the current *Orders* table:

OrderID	CustomerName	Items
101	Juan Dela Cruz	Burger, Fries, Coke
102	Ana Reyes	Burger, Fries
103	Pedro Santos	Fries, Coke, Nuggets

1NF version:

OrderID	CustomerName	Items
101	Juan Dela Cruz	Burger
101	Juan Dela Cruz	Fries
101	Juan Dela Cruz	Coke
102	Ana Reyes	Burger
102	Ana Reyes	Fries
103	Pedro Santos	Fries
103	Pedro Santos	Coke
103	Pedro Santos	Nuggets

1. Rewrite this table so **each cell contains only one value**.
2. Keep OrderID and CustomerName, but make sure each row contains only **one item**.

Stage 2 – Split the Kitchen → Second Normal Form (2NF)

Goal: Remove partial dependencies when using a composite key.

Instructions:

1. Consider the *Order Details* table below (OrderID + ProductID is the primary key):

OrderID	ProductID	ProductName	Price
101	P01	Burger	75
101	P02	Fries	45
102	P01	Burger	75
103	P02	Fries	45

2NF Version:

Product:

ProductID	ProductName
P01	Burger
P02	Fries

Order Details:

OrderID	Product Name	Price
101	Burger	75
101	Fries	45
102	Burger	75
103	Fries	45

1. Identify which data depends only on **ProductID** and should be moved to another table.
2. Create two separate tables: **Order Details** and **Product**.

Stage 3 – Separate the Managers → Third Normal Form (3NF)

Goal: Remove transitive dependencies.

Instructions:

1. Look at the *Employee* table:

EmployeeID	EmployeeName	DepartmentID	DepartmentName
E01	Carla Cruz	D01	Kitchen
E02	Mark Reyes	D02	Service
E03	Ana Santos	D01	Kitchen

3NF Version:

Employee:

EmployeeID	EmployeeName	DepartmentID
E01	Carla Cruz	D01
E02	Mark Reyes	D02
E03	Ana Santos	D01

Department:

DepartmentID	DepartmentName
D01	Kitchen
D02	Service

1. Which column depends on **DepartmentID** instead of **EmployeeID**?
2. Split the data into two tables: **Employee** and **Department**.

Stage 4 – BCNF Burger Rules

Goal: Ensure every determinant is a candidate key.

Instructions:

1. Review this *Branch Manager* table:

ManagerID	BranchID	ManagerName	BranchLocation
M01	B01	John Cruz	Manila
M02	B01	Maria Reyes	Manila
M03	B02	Pedro Juan	Cebu

BCNF Version:

ManagerID	ManagerName
M01	John Cruz
M02	Maria Reyes
M03	Pedro Juan

BranchID	BranchLocation
B01	Manila
B02	Cebu

1. Identify dependency issues between ManagerID, BranchID, and BranchLocation.
2. Suggest how to split the table to follow BCNF.

Stage 5 – No Multi-Valued Combos → Fourth Normal Form (4NF)

Goal: Remove multi-valued dependencies.

Instructions:

1. Review this *Customer Preferences* table:

CustomerID	FavoriteDrink	FavoriteSnack
C01	Coke	Fries
C01	Sprite	Burger
C02	Iced Tea	Nuggets

4NF Version:

Drinks:

Snacks:

CustomerID	FavoriteDrink
C01	Coke
C01	Sprite
C02	Iced Tea

CustomerID	FavoriteSnack
C01	Fries
C01	Burger
C02	Nuggets

1. Notice that drinks and snacks are independent lists.
2. Break them into **two separate tables** for drinks and snacks.

Stage 6 – Perfect Combo → Fifth Normal Form (5NF)

Goal: Remove unnecessary join dependencies.

Instructions:

1. Review this *Combo Deals* table:

ComboID	BurgerType	DrinkType	SideType
CMB01	Cheeseburger	Coke	Fries
CMB02	Chicken Burger	Sprite	Nuggets

5NF Version:.

Burgers: Drinks: Sides:

BurgerID	BurgerType
B01	Cheeseburger
B02	Chicken Burger

DrinkID	DrinkType
D01	Coke
D02	Sprite

SideID	SideType
S01	Fries
S02	Nuggets

1. Think about how to split this into three separate tables for Burgers, Drinks, and Sides, so combos can be created dynamically.

REFLECTION

1. Which normalization stage do you think makes the biggest improvement in a fast-food database?
2. How can normalization help in preventing errors in the FastBite ordering system?
3. Can too much normalization be a bad thing? Why?
